Automatic solver for non-linear partial differential equations with implicit local laws. Application to unilateral contact.

Arnaud Lejeune, Hakim Boudaoud, Michel Potier-Ferry, Isabelle Charpentier, Hamid Zahrou

February 12, 2014

Abstract

In general, nonlinear continuum mechanics combine global balance equations and local constitutive laws. In this work, frictionless contact between a rigid tool and a thin elastic shell is considered. This class of boundary value problems involves two nonlinear algebraic laws: the first one gives explicitly the stress field as a function of the strain throughout the continuum part, while the second one is a nonlinear equation relating the contact forces and the displacement at the boundary.Given the fact that classical computational approaches sometimes require significant effort in implementation of complex nonlinear problems, a computation technique based on Automatic Differentiation of constitutive laws is presented in this paper. The procedure enables to compute automatically the higher-order derivatives of these constitutive laws and thereafter to define Taylor series that are the basis of the continuation technique called Asymptotic Numerical Method. The algorithm is about the same with an explicit or an implicit constitutive relation. In the modeling of forming processes, many tool shapes can be encountered. The presented computational technique permits an easy implementation of these complex surfaces, for instance in a finite element code: the user is only required to define the tool geometry and the computer is able to get the higher-order derivatives.

Asymptotic Numerical Method (ANM); contact mechanics; shell element; implicit surface

1 Introduction

Many industrial processes require the mathematical description of complex surfaces. This work is a part of a research project dealing with the simulation of a sheet metal conveying, with application to steel fabrication plants, especially continuous annealing, levelling or galvanization. In the conveying process, the rolls are imperfect cylinders with a non smooth surface due to wear or to improve the control of process. As a consequence of this non cylindrical shape, wrinkling phenomena can occur, leading to difficulties in the management of the plant [1] [2]. Unilateral contact involves a non smooth relation between the contact forces and the distance between the two bodies. The numerical techniques to solve unilateral contact are often based on Lagrange multipliers [3], augmented Lagrangian [4] [5] and regularization of the non smooth function like in penalty method [6] [7] [8]. The tool shape may have significant impact on the deformation and the stability of the sheet [1]. Hence, it is important to describe the complex surfaces with accuracy. Bézier, B-spline and NURBS are among the most popular in the field of simulation of forming processes, see for instance [9].

This paper is devoted to a generic modelling approach of unilateral contact, so that the results of the paper could be applied to any structural and processing cases involving contact. A new procedure involving Automatic Differentiation (AD) and Asymptotic Numerical Method (ANM), to account for complex tool shape, is presented. The Asymptotic Numerical Method is an efficient path following technique for the solution of strongly non-linear partial differential equations. For instance it has been applied to problems of unilateral contact [10] [11] [12] [13], plasticity [14] [15], viscoplasticity [16] and to problem coupling several non-linearities [16] [17].

The appearances of wrinkles in strip conveying [1] or of flatness defects after rolling process of thin sheets [2] are instability phenomena induced by residual stresses due to fabrication processes. Because ANM has proved to be a very efficient continuation algorithm in order to predict all sorts of buckling phenomena, see for instance [18], it is a good candidate to design software dedicated to the simulation of strip conveying. Up to recently, ANM was not well adapted to the description of generic tool shapes and to account for plasticity and friction. The present work focuses on the first point with the help of Automatic Differentiation while the treatment of plasticity and friction is presented in companion papers [14] [19].

The ANM algorithm does not differ significantly from the one presented in [13]. The contact law is regularized by a smooth function and the main unknowns are the displacement and the contact forces as in many contact algorithms. Up to now within ANM, the tool surfaces have been represented by simple surfaces as planes or cylinders [10] [12] [13]. The main goal of this paper is to establish an efficient and generic methodology to account for complex surfaces. Then, AD is used to compute the higher-order derivatives of constitutive laws. This allows for the automation of the implementation of various complex tool shapes, tool motions or gap-pressure relations. There have been many works about Automatic Differentiation [20] [21] and about its applications to partial differential equations using or not finite element method (see for instance [22], [23], [24], [25], [26],...), but only few papers have described coupling of AD and ANM [27] [28] [21] [29].

The layout of the paper is as follows: Section 2 presents the higher-order differentiation of some local constitutive laws for nonlinear PDE problems and

Section 3 introduces the treatment of the global problem with unilateral contact. Section 4 illustrates the resolution of a global contact problem with complex contact surfaces. Some perspectives are proposed as a conclusion.

2 Higher-order differentiation of local constitutive laws

In Asymptotic Numerical Method, the unknowns U of the non-linear problem are written in the form of truncated Taylor series in the vicinity of a starting point U_0 :

$$a \in \mathbb{R} \rightarrow U(a) = U_0 + \sum_{k=1}^{K} a^k U_k. \tag{1}$$

Hence the computation of a solution branch is equivalent to the computation of the coefficients of the expansion that are obviously related to the higher-order derivatives of the function :

$$U_k = \frac{1}{k!} \frac{\partial^k U}{\partial a^k}(0). \tag{2}$$

Suppose now that two variables of the problem are related by an algebraic relation such as the constitutive law of non-linear elasticity that connects the strain and the stress. It is well known within ANM [13] [14] that such a law leads to a constitutive law at order k: the higher-order derivative of the stress is a function of derivatives of the strain. The goal of this part is to show how this constitutive law at order k could be implemented automatically using Automatic Differentiation, even if the starting law is complex. As usual in continuum mechanics, the full problem associates global balance equations and local constitutive laws. In this part, we focus only on the treatment of constiitutive laws. The full problem will be discussed in Part 3.

2.1 Algebraic constitutive laws

2.1.1 Material modeling

Let us recall that the deformation gradient tensor F is related to the displacement gradient tensor $\theta = \nabla_x u$ by $F = I_2 + \theta$ where I_2 is the second order identity tensor. The Green-Lagrange strain tensor γ is defined by

$$\gamma = \frac{1}{2}\left({}^T F F - I_2\right) = \frac{1}{2}\left({}^T\theta + \theta + {}^T\theta\theta\right). \tag{3}$$

Let us consider the first and second Piola-Kirchhoff tensors, respectively denoted T and S related through $T = (I_2 + \theta) S$.

In elasticity framework, S and T are related to the Green Lagrange second order strain tensor by the fourth order elastic tensor L:

$$\begin{aligned} T &= (I_2 + \theta) S, & (4) \\ S &= L : \gamma, & (5) \\ \gamma &= \frac{1}{2} \left({}^T\theta + \theta +^T \theta\theta \right). & (6) \end{aligned}$$

The computation of the derivatives of the constitutive laws (4)-(6) is well known and is accounted by straightforward formulae :

$$\begin{aligned} T_k &= (I_2 + \theta_0) S_k + \sum_{i=1}^{k} \theta_i S_{k-i}, & (7) \\ S_k &= L : \gamma_k, & (8) \\ \gamma_k &= \frac{1}{2} \left(\left({}^T\theta_k + \theta_k \right) + \sum_{i=0}^{k} {}^T\theta_i \theta_{k-i} \right). & (9) \end{aligned}$$

AD avoids the implementation of the derivative formulae (7)-(9) : the user has only to implement the basic laws (4)-(6), the derivatives being evaluated by the use of operator overloading technique. This is a direct mode of automatic differentiation.

2.1.2 Contact modeling : the geometrical part

Projection algorithm consists in computing the distance h and the normal vector n between the slave point and the rigid master surface. The normal vector is defined on the rigid master surface. By denoting X^0 the initial coordinate of the observed slave point and u its displacement field, the slave node is located at the coordinate $X = X^0 + u$. Forthcoming subsections briefly describe different kinds of functions used to express the projection function linking the displacement field u to the projection variables h and n. This function depends on the master surface geometric model. Papers on sheet metal forming are related to Bezier, NURBs or analytic description [9].

Explicit Projection This case is the most simple. The projection variables h, n are explicitly related to the displacement field u:

$$proj\,(h, n, u) =^T \{h, n\} - f\,(u) = 0. \qquad (10)$$

with f a given function depending on the surface definition. Higher-order differentiation is of no problem and treated as the elastic constitutive equation.

Implicit Projection with non-parametric surface The master surface is described by geometrical variables (for example radius and center coordinates for a sphere). The relationship between the normal vector, the distance and the displacement field is given by the following implicit function :

$$proj\,(h, n, u) = 0. \tag{11}$$

For example, let us consider a perfect cylinder, given by its center point C, its radius value r with an axis parallel to (Oy). The projection on this cylinder is given by:

$$proj\,(u, h, n) = \begin{cases} (h+r)n_x - x^0 + C_x - u_x = 0 \\ n_y = 0 \\ (h+r)n_z - z^0 + C_z - u_z = 0 \\ \langle n, n \rangle - 1 = 0 \end{cases} \tag{12}$$

Implicit Projection with parametric surface Another way to describe an obstacle is to use parametric surfaces such as Bezier surfaces for instance. Let us denote by $S\,(s)$ the parametric surface where we suppose that s is in a bounded domain of dimension L.

To compute the variables h and n, one has to know the coordinate of the point projected from the continuum media (deformable solid) onto the parametric surface. Then finding the distance h and the normal n is equivalent to solving the following problem:

Find (h, n, s) such that:

$$proj(h, n, s, u) = \begin{cases} X^0 + u - S\,(s) - hn & = & 0 \\ \langle n, n \rangle - 1 & = & 0 \\ \left\langle n, \frac{\partial S}{\partial s}\,(s) \right\rangle & = & 0 \end{cases} \tag{13}$$

So considering parametric surfaces leads to increase of the dimension of the local projection problem by the dimension of parametric variables.

2.1.3 Contact modeling : the mechanical part

In this paper, we only discuss frictionless contact. The contact pressure is assumed to be normal to the two bodies:

$$R = R^N n. \tag{14}$$

The Asymptotic Numerical Method (ANM) requires smooth constitutive laws. As proposed in [13] the hard contact law ($R^N \geq 0$, $h \geq 0$, $R^N h = 0$) may be replaced by a smooth relation between the normal pressure R^N and the gap h, following:

$$R^N(\frac{R^N}{\alpha} + h) = \eta(\Delta - h). \tag{15}$$

In (15), Δ is an initial gap ($\Delta > 0$) and α and η are two parameters. The given number η has the dimension of a pressure and it is assumed to be small. For a very small η, the relation (15) tends to a piecewise linear function $R^N \approx 0$ if $h > 0$, $R^N = \alpha(-h)$ if $h < 0$. Hence the parameter α is a kind of contact stiffness parameter and it has to be large. If $1/\alpha = 0$ the penetration is prevented and the relation (15) leads to a barrier function as in [30]. The barrier contact law ($1/\alpha = 0$) has been used in many papers about ANM, for instance [31] [13]. By combining (15) with the projection equations of subsection 2.1.2, the contact normal pressure R^N can be considered as an implicit function of the displacement u. In the same way, the pressure vector R is considered as a function of R^N and u. In the next part, we present a general numerical procedure to compute the higher-order derivatives of these functions $u \rightarrow R^N$ and $(R^N, u) \rightarrow R$.

2.2 DIAMANT approach applied to local laws

Continuum mechanics deal with non-linear problems implying complex local non-linear laws. Such problems can be solved using the ANM through the computation of higher-order derivatives in a particular direction. The approach, named DIAMANT, coupling Automatic Differentiation (AD) and ANM has been proposed in [32].

2.2.1 DIAMANT approach

As noted previously in the introduction of part 2, unknowns are expanded in Taylor series versus a scalar path parameter. ANM algorithm consists in finding each Taylor coefficient. These terms are related to the higher-order derivative with respect to the path parameter. The non-linear problem is also expanded in Taylor series leading to a system of linear equations at each order. Expressing higher-order equations with respect to Taylor coefficients could be difficult and error-prone. Then coupling ANM with AD (namely the DIAMANT approach) allows to automate Taylor series terms in a specific direction using well-known recurrence formula.

Recently, DIAMANT approach has been proposed and applied to several kinds of academic problems [32], [21], [27]. Right-hand side terms and tangent matrix occurring in ANM are obtained by application of AD technique. This technique has been originally proposed in [20], in which the package ADOL-C for automatic differentiation [33] is employed. It is quite related to several works about solving ODE problems (see for example [34]) or to Tadiff software [35].

Thus we retrieve the same differences between our approach and the use of ADOL-C as noted in [34]. Moreover, [32] recalls that Taylor series computation does not need higher-order derivatives in all directions. In this sense, it is more simple than classical higher-order AD tools.

To illustrate this approach, let us recall the automatic computation of a variable defined by an explicit function itself expressed as a sequence of elementary operations. Let y be a variable defined as:

$$y = f\left(x\right). \tag{16}$$

Using Faà di Bruno's formula as in [21], one could demonstrate that the latter relation is linear with respect to x_k and therefore can be written as:

$$\begin{aligned} y_k &= \frac{\partial f}{\partial x}(x_0)x_k + f_k^{nl}\left(x_{k-1},...,x_1,x_0\right), \\ f_k^{nl} &= f_k\left(0,x_{k-1},...,x_1,x_0\right). \end{aligned}$$

In the latter formula, two main quantities appear that are fundamental within ANM; the tangent matrix $\frac{\partial f}{\partial x}(x_0)$ and the right-hand side f_k^{nl} that can be seen as a higher-order derivative of the function $f(x)$. Hence, implementing ANM is equivalent to computing higher-order derivatives of function. Derivatives up to the order k of y could be computed with AD by using operator overloading. A sequence of computation could be:

$$\begin{aligned} y_0 &= f\left(x_0\right), \\ y_1 &= f_1\left(x_1,x_0\right), \\ &\vdots \\ y_K &= f_K\left(x_K,\ldots,x_1,x_0\right). \end{aligned}$$

Principle of computation of k^{th} derivative of f function by AD is based on the differentiation chain rule. The analytical function f depends on x from a compound of elementary arithmetic operators. As the higher-order differentiation of these elementary operators is well known, applying the function f onto differentiation of variable would lead to the computation of higher-order derivatives f_k of the function f. As an example, consider this arbitrary function:

$$f\left(x\right) = x^2 + e^{3x}. \tag{17}$$

Naïve successive computation operations could be described by pseudo-code given in Algorithm 1.

AD by operator overloading described in subsection 2.2.3 would lead to the computation of derivatives using recurrence formulae. For instance a differentiation code specific of the function described by the Algorithm 1 is given in Algorithm 2. The generic AD code combines elementary operations that corresponds to each line of Algorithm 2.

Algorithm 1 Successive operations to compute $f\left(x\right) = x^2 + e^{3x}$

```
$u = x * x$
$v = 3 * x$
$w = e^v$
$f = u + w$
```

Algorithm 2 Successive operations to compute derivatives of $f\left(x\right) = x^2 + e^{3x}$

```
$u_k = \sum_{l=0}^{k} x_l * x_{k-l}$
$v_k = 3 * x_k$
$w_k = \frac{1}{k} \sum_{l=0}^{k-1} \left(k - l\right) w_l v_{k-l}$
$f_k = u_k + w_k$
```

2.2.2 DIAMANT approach extended to local implicit functions

Until now, problems treated with DIAMANT approach only deal with local explicit functions. A difficulty arises when considering local implicit equations for the contact projection especially for complex contact surfaces. Then we would like to extend DIAMANT approach to treat more complex industrial problems modelled with continuum mechanics and complex local laws. This extension is turned into a generic approach to solve analytical non-linear problems with complex local laws. The originality of our work consists in mixing non-linear global equations and computation of local variables from local implicit and explicit non-linear laws.

Compared to explicit function, dealing with an implicit function is slightly different. Explicit function is computed with a formula that is composed of its variables and elementary operations. On the contrary, an implicit function needs the use of an algorithm to express one variable with respect to the other one. Assume y is related to x by such a function (with all the needed properties for the sequence):

$$g\left(x, y\right) = 0. \tag{18}$$

One starts from a known solution (x_0,y_0): in practice, this starting point will be the endpoint of the previous step. The derivative expression of the order $k > 0$ of equation (18) is given by:

$$g_k = \frac{\partial g}{\partial x} x_k + \frac{\partial g}{\partial y} y_k + g_k^{nl}\left(x_{k-1}, \ldots, x_1, x_0; y_{k-1}, \ldots, y_1, y_0\right) = 0. \tag{19}$$

As the former derivatives are linear with respect to x_k, y_k, the relation (19)

Algorithm 3 Computation of a term defined by an implicit function

```
Require: $\mathbf{x}_0$, $\mathbf{y}_0$ / $\mathbf{g}\left(\mathbf{x}_0, \mathbf{y}_0\right) = 0$
Compute $\mathbf{k}_T = -\left(\frac{\partial \mathbf{g}}{\partial \mathbf{y}}\right)^{-1}$
ORDER 1
Compute $\tilde{g}_1 = \mathbf{g}_1|_{\mathbf{y}_1=0}$
Compute $\mathbf{y}_1 = \mathbf{k}_T \tilde{g}_1$
ORDER $k > 1$
for $k = 2$ to $K$ do
    Compute $\tilde{g}_k = \mathbf{g}_k|_{\mathbf{y}_k=0}$
    Compute $\mathbf{y}_k = \mathbf{k}_T \tilde{g}_k$
end for
return $\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_k$
```

can be inverted by classical algebra:

$$y_k = -\left(\frac{\partial g}{\partial y}\right)^{-1} \left(\frac{\partial g}{\partial x} x_k + g_k^{nl}\left(x_{k-1}, \ldots, x_1, x_0; y_{k-1}, \ldots, y_1, y_0\right)\right). \tag{20}$$

To automate the process, we note that the term on the right hand side corresponds to the computation of g_k restricted to $y_k = 0$. Consequently, a driver has been proposed to compute implicit function derivatives up to the order k following Algorithm 3.

When the implicit function is a local one included in a more general process, computation time is reduced by storing each result at a given order and just computing the given order. Note that this computation of the higher-order derivative can be seen as a specific higher-order extension of the Post-Differentiation defined in [36]. They conclude that, in the first order derivative case, the Post-Differentiation approach seems to be less expensive than a Full Differentiation one in terms of cost of iterations.

2.2.3 AD with Operator Overloading technique

This section is a brief recall of the AD technique in forward mode applied to the computation of higher-order Taylor series terms.

Following the definition of [37], AD could be viewed as the way to generate a code that computes the derivative of a function itself given by a code. A natural approach is to use the forward mode by applying operator overloading. This is notably well-suited to C++ language [38]. This language allows to overload classical operators in a simple way [39]. Notice that AD by operator overloading has already been implemented since a long time in other classical languages [40]. Interested readers can refer to [41] for a good review of AD techniques. But in most cases, works are done for low order derivatives (first or second order). We are here concerned with higher-order derivatives in direct mode as in [42] but we only need the computation of the derivatives in a specific direction.

From a programing point of view, higher-order differentiation by operator overloading may be realized using template facilities of C++ language. First, the function to be derived is implemented and tested in the continuous domain. Then the template parameter is changed into the higher-order derivative one. Consequently, one single implementation allows the computation of continuous residuals and derivatives. A particuliar procedure is applied in the case of local implicit functions due to the different nature between the continuous function (implicit) and the derivative function (explicit).

As an illustration a naïve templated implementation of the example function $f(x) = x^2 + e^{3x}$ presented in subsection 2.2.1 is given in the following

```
template < class T_Scalaire>
T_Scalaire fonction_f(T_Scalaire x)
{
  return (x*x + exp(3*x));
}
```

It is then possible to compute the function value in the continuous domain

```
double computeValue(double x)
{
  double y;
  y = fonction_f< double >(x)
}
```

and its derivatives:

```
// We here suppose that T_Diff is the class describing derivative types
```

```
T_Diff computeDerivatives(T_Diff x)
{
  T_Diff y;
  y = fonction_f< T_Diff >(x)
}
```

One could note that switching between scalar and derivative computation results in changing variable type. As recurrence formulae are described in the *T_Diff* class with operator overloading ability, computing the scalar function would then apply the Algorithm 1 successive operations while computing the derivative function woud refer to Algorithm 2 operations.

2.3 Higher-order derivative of the projection equations

2.3.1 Explicit Projection

This is the simplest case. Following AD well-known technique, derivative of the local law at the order k is given by:

$$ {}^{T}\left\{h_k, n_k\right\} = f_k\left(u\right). \tag{21}$$

with f_k the derivative at the order k of the given function f depending on the surface definition. This derivative could be expressed in term of linear and non-linear part (f_k^{nl}) as:

$$f_k\left(u\right) = \frac{\partial f}{\partial u} u_k + f_k^{nl}\left(u_0, u_1, \ldots, u_{k-1}\right). \tag{22}$$

Using AD technique results in implementing function f and letting AD compute the Taylor series terms (linear and non linear).

2.3.2 Implicit Projection with non-parametric surface

Dealing with an implicit local projection law leads to solving the following problem

$$\begin{array}{c}\text{Knowing } u \text{, find } h\left(u\right) \text{ and } n\left(u\right) \text{ such as}\\ proj\left(h, n, u\right) = 0.\end{array} \tag{23}$$

Using subsection 2.2.2 and denoting $\Pi = (h, n)$, one could apply the Algorithm 3 so that the non-linear local laws are replaced by linear ones:

$$\Pi_k = -\left[\tfrac{\partial proj}{\partial \Pi}\left(\Pi_0, u_0\right)\right]^{-1}\left(\tfrac{\partial proj}{\partial u}\left(\Pi_0, u_0\right) u_k + proj_k^{nl}\left(u_0, \ldots, u_{k-1}; \Pi_0, \ldots, \Pi_{k-1}\right)\right). \tag{24}$$

2.3.3 Implicit Projection with parametric surface

This is the same as implicit projection with non-parametric surface except that the problem dimension is increased by the number of parameters defining the parametric surface:

$$\begin{array}{c}\text{Knowing } u\text{, find } h\left(u\right)\text{, } n\left(u\right) \text{ and } s\left(u\right) \text{ such as}\\ proj\left(h,n,s,u\right)=0.\end{array} \tag{25}$$

Denoting $\tilde{\Pi}=(h,n,s)$ yields a formula similar to (24) :

$$\tilde{\Pi}_k=-\left[\frac{\partial proj}{\partial\tilde{\Pi}}\left(\tilde{\Pi}_0,u_0\right)\right]^{-1}\left(\frac{\partial proj}{\partial u}\left(\tilde{\Pi}_0,u_0\right)u_k+proj_k^{nl}\left(u_0,\ldots,u_{k-1};\tilde{\Pi}_0,\ldots,\tilde{\Pi}_{k-1}\right)\right). \tag{26}$$

3 Higher-order differentiation of global problems with unilateral contact

The library described in part 2 is applied in solving the finite element modeling of unilateral contact problems between a rigid tool and a deformable elastic body. The constitutive assumptions are those presented in section 2.1: the geometrical quantities (strain, gap between the two bodies) are defined exactly. The stress-strain relation is linear, the contact pressure is defined by a regularized version of the hard frictionless contact law. These constitutive assumptions are accounted by three local functions. The elasticity laws leads to an explicit function $\theta=\nabla u\rightarrow T$, the smooth contact law requires two functions: the first one defines the normal pressure as a function of the displacement ($u\rightarrow R^N$) and the second one defines the pressure vector as a function of the displacement and of the normal pressure $(R^N,u)\rightarrow R$. To implement the ANM in a finite element code, one needs only linear relations deduced from the higher-order derivatives of those functions: these higher-order constitutive laws are computed according to the automatic differentiation procedures presented in section 2.2. The constitutive functions of the present paper have been chosen for simplicity. The contact law is more or less generic in the frictionless framework, but the stress-strain relation is very specific. The present approach could be extended for instance to hyperelasticity, which has been solved by ANM in few papers [43] [44] [45], because hyperelasticity leads also to smooth algebraic functions $\theta\rightarrow T(\theta)$. Alternative constitutive laws like plasticity [14] or friction [19] implies Differential Algebraic Equations (DAE) that requires a slightly different treatment, but the DIAMANT approach can also be used in case of DAEs.

3.1 Unilateral contact with an elastic 3D body

The deformable body is discretized with 3D elements. A mixed formulation in displacement/contact pressure is proposed. The balance of momentum equation and the boundary conditions are accounted to define a well posed boundary value problem. The equivalent weak form is the virtual work equation. Most of the computer codes involving contact mechanics are based on a mixed approach: the discrete unknowns are the displacement and the contact pressure. Within the ANM, this Lagrange multiplier method seems better than a pure displacement method [13]. Hence the full elastic-contact problems can be summarized as follows:

$$\text{Find } \left(u, R^N, \lambda\right) \text{ such as :}$$

$$\left\{ \begin{array}{l} \displaystyle\int_\Omega \hat{T}.\nabla \delta u d\Omega - \int_{\partial\Omega} \hat{R}(R^N, u).\delta u d\Gamma - \lambda P_e(\delta u) = 0, \\ \\ R^N = \hat{R}^N(u). \end{array} \right. \tag{27}$$

where the constitutive functions $T = \hat{T}(\nabla u)$, $R = \hat{R}(R^N, u)$ and $R^N = \hat{R}^N(u)$ have been defined in part 2. To compute the series $u\,(a)$, $R^N\,(a)$ and $\lambda\,(a)$, one needs the derivatives at any order k of their constitutive functions. They are in the form :

$$T_k = \frac{\partial \hat{T}}{\partial \theta}\theta_k + T_k^{nl} = \frac{\partial \hat{T}}{\partial \theta}\nabla u_k + T_k^{nl}, \tag{28}$$

$$R_k = n_0.R_k^N + \frac{\partial \hat{R}}{\partial u}.u_k + R_k^{nl}, \tag{29}$$

$$R_k^N = \frac{\partial \hat{R}^N}{\partial u}.u_k + {R^N}_k^{nl}. \tag{30}$$

In the formula (28), $\frac{\partial \hat{T}}{\partial \theta}$ is nothing but the usual tangent stiffness matrix at the starting point of the series. The details of the functions $T_k\,(\theta_k, \theta_{k-1}, \ldots, \theta_1, \theta_0)$ have been given in formulas (7), (8) and (9).
The latter formulae (29) and (30) combine the order k derivatives of the scalar functions $R^N\,(h)$ given in (15) and the k^{th}-derivatives of the functions $h\,(u)$ and $n\,(u)$ computed like in subsection 2.3.3.

Finally, combination of (28), (29) and (30) and the obvious k^{th}-derivative of the problem (27) leads, at a given order k, to the linear set of equations satisfied by the mixed unknown $\left(u_k, R_k^N\right)$ and the k^{th} component λ_k of the load parameter:

$$\text{Find } \left(u_k, R_k^N, \lambda_k\right) \text{ such as :}$$

$$\left\{\begin{array}{ll} \displaystyle\int_\Omega \nabla\delta u.\frac{\partial\hat{T}}{\partial\theta}.\nabla u_k d\Omega & -\int_{\partial\Omega}\delta u.\frac{\partial\hat{R}}{\partial u}.u_k d\Gamma - \int_{\partial\Omega^c}\delta u.n_0 R_k^N d\Gamma^c - \lambda_k P_e(\delta u) \\ & = \int_{\partial\Omega^c}\delta u.R_k^{nl} d\Gamma^c - \int_\Omega \nabla\delta u.T_k^{nl} d\Omega \\ -\frac{\partial\hat{R}^N}{\partial u}u_k + \hat{R}_k^N & = {R^N}_k^{nl} \end{array}\right. \tag{31}$$

In this form, the linear problem (31) does not differ significantly from the formulations presented in [13]. So the discretization is quite the same. The main difference lies in the computation method of the right hand sides R_k^{nl}, T_k^{nl} and ${R^N}_k^{nl}$ and in the application to any tool shape described with explicit or implicit functions.

3.2 Unilateral contact with an elastic shell

In view of application to sheet metal conveying process simulation, the body has to be discretized by a shell element. As in [18], a three-dimensional shell formulation proposed by Buchter-Ramm-Roehl in [46] is used. An advantage of this element is its strong similarity with volumetric elements and moreover the three-dimensional constitutive law could be implemented directly.

Hence the mathematical formulation and the discretized technique would be quite similar to those of the three dimensional case. The main difference is the introduction of a new strain variable, denoted by $\tilde{\gamma}$ to account for a linear variation of γ_{33} through the thickness. This extra variable is incorporated via the Enhanced Assumed Strain (EAS) concept introduced in [47].

The elastic constitutive law (4), (5), (6) is modified. It is an explicit function $(\theta = \nabla u, \tilde{\gamma}) \to (T, S)$ that is written as:

$$\begin{aligned} T &= (I_2 + \theta)\, S && (32) \\ S &= L : (\gamma^c(\theta) + \tilde{\gamma}), && (33) \\ \gamma^c(\theta) &= \frac{1}{2}\left({}^T\theta + \theta + {}^T\theta\theta\right). && (34) \end{aligned}$$

where (32), $\gamma^c(\theta)$ is the conventional Green-Lagrange strain.

As a counterpart, a new equation is introduced that is an orthogonality condition between the second Piola-Kirchhoff stress tensor and the enhanced part of the strain $\tilde{\gamma}$.

The full contact problem (27) is modified as follows:

$$\text{Find } \left(u, R^N, \lambda\right) \text{ such as :}$$

$$\left\{\begin{array}{l} \displaystyle\int_\Omega \hat{T}(u,\tilde{\gamma}) : \nabla\delta u d\Omega - \int_{\partial\Omega}\hat{R}(R^N, u).\delta u d\Gamma - \lambda P_e(\delta u) = 0 \\ R^N = \hat{R}^N(u) \\ \displaystyle\int_\Omega \hat{S}(u,\tilde{\gamma}) : \delta\tilde{\gamma} d\Omega = 0 \end{array}\right. \tag{35}$$

The higher-order differentiation of (35) is straightforward and quite similar to the volumetric case, the only difference being the account of the last equation leading to an affine relationship between ∇u_k and $\tilde{\gamma}_k$.

3.3 Finite element discretization, ANM algorithm

The key point of this work is the automatic computation of the linear problem defining the derivatives at any order which has been presented up to now. The remainder of the method is very classical: spatial discretization using the Finite Element Method and continuation technique within the ANM.

In the application discussed in the following section, we limit ourselves to the contact between a thin shell and a rigid tool. The deformable body is discretized using 8-nodes quadrilateral shell elements with quadratic shape functions introduced in [46] and based on the EAS concept [47].

The implementation in a finite element framework is the same as in [18]. The contact pressures are represented by forces at each node as in [13].

The Finite Element discretization and the assembly step are not impacted by the AD technique that is only applied at the level of the elements. For instance, the four degrees of freedom accounting for the additional strain $\tilde{\gamma}$ are eliminated at this level using a condensation technique.

Note also that the continuation algorithm is really independent of differentiation method. Then it is not affected by the AD technique.

Once the series are computed, the length of the solution branch is defined by an estimate of the convergence radius of the series, as discussed for instance in [18]:

$$a_{max} = \left(\delta \frac{|||u_1|||}{|||u_K|||} \right)^{\frac{1}{K-1}} . \qquad (36)$$

where δ is a user-defined accuracy parameter (typically $\delta = 10^{-3}$ or $\delta = 10^{-6}$).

Next another computation of series is performed from the step end defined using a_{max} and so on.

4 Numerical applications

4.1 Two numerical tests

4.1.1 Numerical modelling

Two numerical tests of contact between a thin plate and a rigid roll are considered. The aim is to show that complex tool shapes can be easily and efficiently modelled with the help of the presented numerical tool software that couples ANM and AD.

Two roll shapes are considered. In the first test, it is a perfect cylinder (Figure 1) and in the second case a tapered roll that combines a perfect cylinder and a cone (Figure 2). The main geometric data are the dimensions of the plate (length = 400 mm, width = 20 mm, thickness = 0.8 mm) and the radius of the roll (r = 50 mm) used in the two cases. According to Figure 2, additional data defining the tapered roll are the flat length (length of the cylindrical part) l_1 = 15 mm, the length l_2 = 5 mm and the angle between the cone and the cylinder $p = 0.1$ rad.

The implementation of the cylindrical roll is done by the implicit representation given in equation (13). In the case of the tapered roll, the geometric equation is given by:

$$proj\,(u,h,n) = \begin{cases} (h+r_p)\,n - X + C - u = 0 \\ r_p = r\left(1 + \frac{tan(p)}{2}\,(y+u_y-l_1)\left(1 + tanh\left(\frac{y+u_y-l_1}{\epsilon}\right)\right)\right) \\ n^2 - 1 = 0 \end{cases} \tag{37}$$

where r, p, l_1 are the parameters described above, C is the center point of the roll and $\epsilon = 23$ a parameter allowing a smooth approximation of the $tanh$ function.

With the equation (37), the non smooth surface of the tapered roll is replaced by a smooth one that differs slightly to the previous one near the vertex. Let us underline that the AD method makes easier modelling of the tapered surface, that requires only the implementation of the three formulae (37). The computation of the Taylor series is done automatically as explained previously.

As the aim of this section is to validate the approach, only 20 elements discretize the plate (Figures 1 and 2).

Concerning the boundary conditions applied on the deformable body (i.e. the plate), the long edges are free. One of the short edges is clamped and the second one is submitted to an uniformly distributed load λF, where λ is the load parameter mentioned in equation (35) and the nominal force F is 100 N. The material is elastic, linear and isotropic with data $E = 220$ MPa and $\nu = 0.3$. The contact law is the regularized law (15) that depends on two numerical parameters: a stiffness parameter $\alpha = 10^3$ MPa.m and a non dimensional regularisation parameter $\eta = 0.1$. They have been chosen in such a way that the regularized contact law is sufficiently close to the hard contact law.

The initial distance between the plate and the tools is 4 mm.

4.1.2 Numerical results

The numerical results obtained by the present method are compared with a well established commercial code (Abaqus) using a hard contact law. Let us mention that the efficiency of Asymptotic Numerical Method for contact problem has been assessed in several previous papers, see for instance [12] [17] and this will not be repeated here.

Few numerical results are presented in Figures 3, 4, 5 and 6. In the case of a cylindrical roll (Figure 3), the vertical displacement on a point submitted to the applied load is plotted versus the applied load λ. One sees many slope discontinuities, due to appearance or disappearance of new nodes in contact. To get the response curve of Figure 3 with an ANM order equals 20 and the accuracy parameter $\delta = 10^{-6}$, 95 ANM steps have been necessary, see [12] or [17]. Note that the number of steps could be reduced by using Padé approximants [31] or by changing the accuracy parameter δ or the regularization parameters, η or α. In Figure 4, the evaluation of three contact forces according to λ is plotted and compared to Abaqus results (reference curves). At the beginning ($\lambda = 0$), there is no contact forces, next the contact occurs at each of these three points, then the nodes take off from the cylinders. The comparison with Abaqus results are in good agreement.

Figures 5 and 6 illustrate the effect of the conical part on the displacement curve between a cylindrical roll and a tapered roll on three points positioned on Figure 2. Comparison between ANM solver and Abaqus (reference curves) again shows good agreement. Looking at the point 3 curve on Figure 6, it is important to notice that the regularisation with a *tanh* function for the ANM solver allows to obtain a good description of the tapered roll surface, notably the intersection between conical and cylindrical parts. At our knowledge, it is the first time that an unilateral contact problem with this kind of non regular obstacle is achieved using the ANM.

4.2 Ability of the solver to simulate strip conveying and other forming processes

4.2.1 Simulation of an elastic buckling problem under traction and unilateral contact with a tapered roll

The rolls are imperfect cylinders with a convex shape. We consider here the process on one tapered roll, as described in Figure 7, using the implicit geometric representation equation (37) and the equilibrium defined by the equation (35). This example is quite similar to the previous one excepted that a tensile load is applied, only half of the plate is modelled and geometric data are similar to those typically encountered in industrial conveying process.

The main geometric data on Figure 7 are the dimensions of the strip (horizontal length L = 1000 mm, width $\frac{l_b}{2}$= 200 mm, thickness = 0.2 mm, table length $\frac{l_t}{2}$ = 100 mm), the radius of the roll r = 250 mm and the angle p = 0.00577

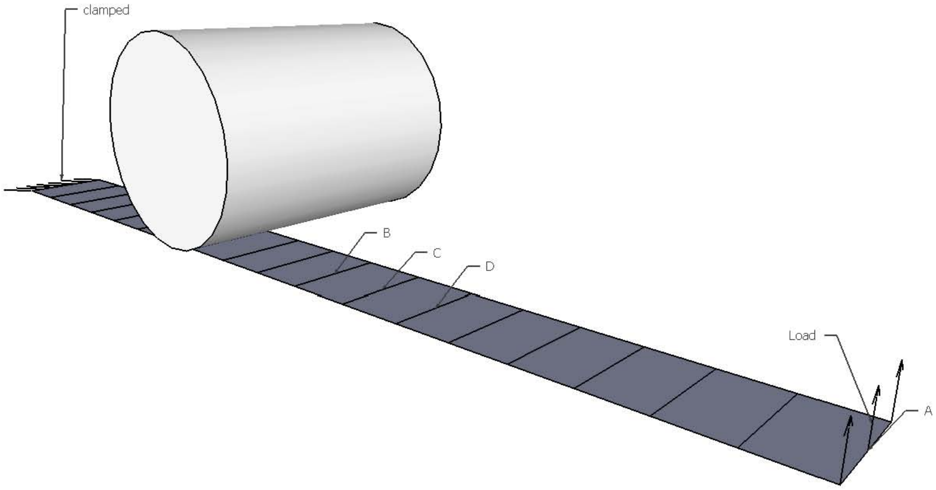

Figure 1: Cylinder/sheet geometry

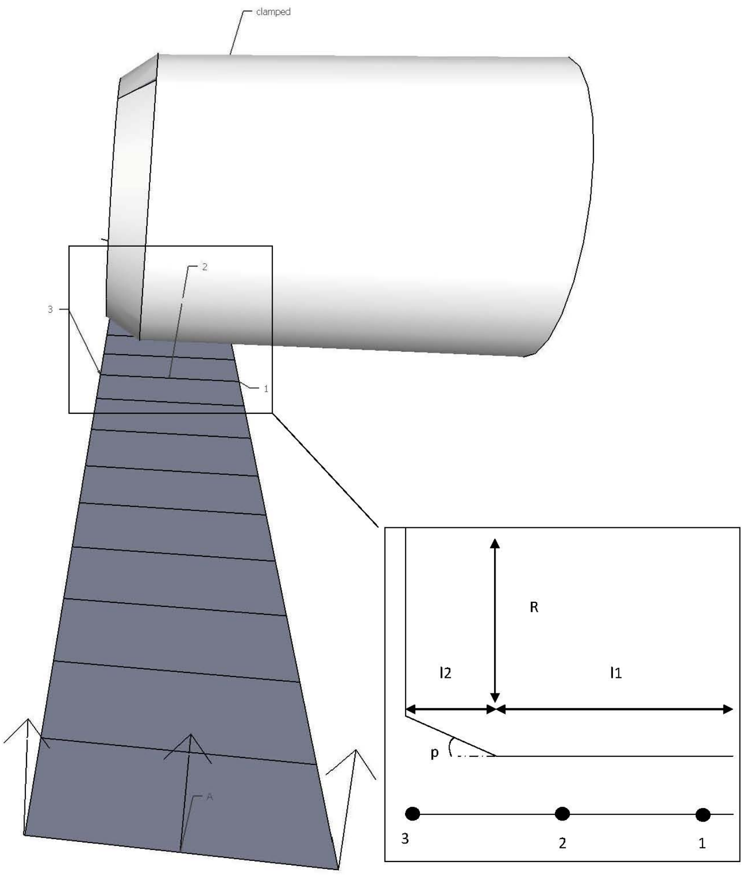

Figure 2: Conical cylinder/sheet geometry

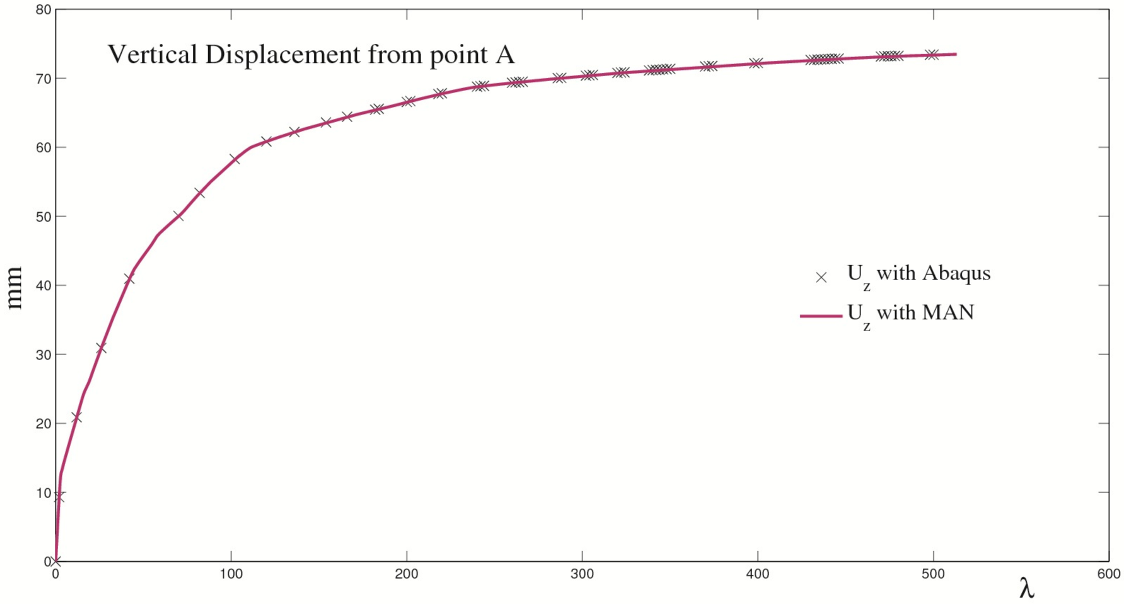

Figure 3: Vertical displacement versus load parameter at point A. Case of the cylindrical roll

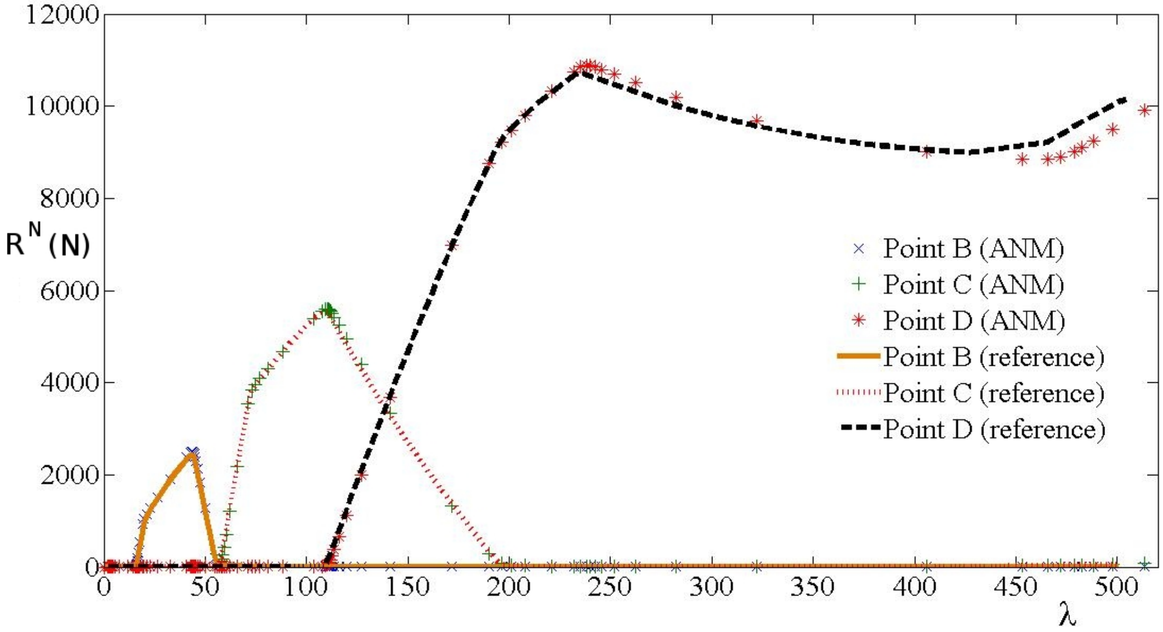

Figure 4: Normal contact reaction versus load parameter at points B, C and D.

Case of the cylindrical roll

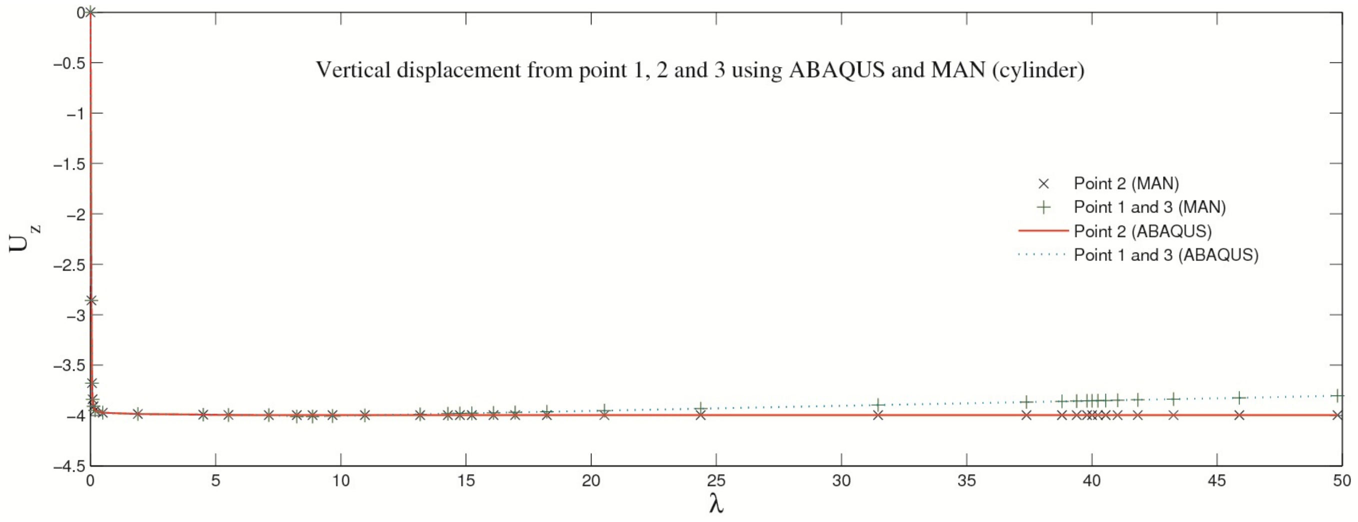

Figure 5: Vertical displacement profile along a perfect cylinder. ANM versus Abaqus results.

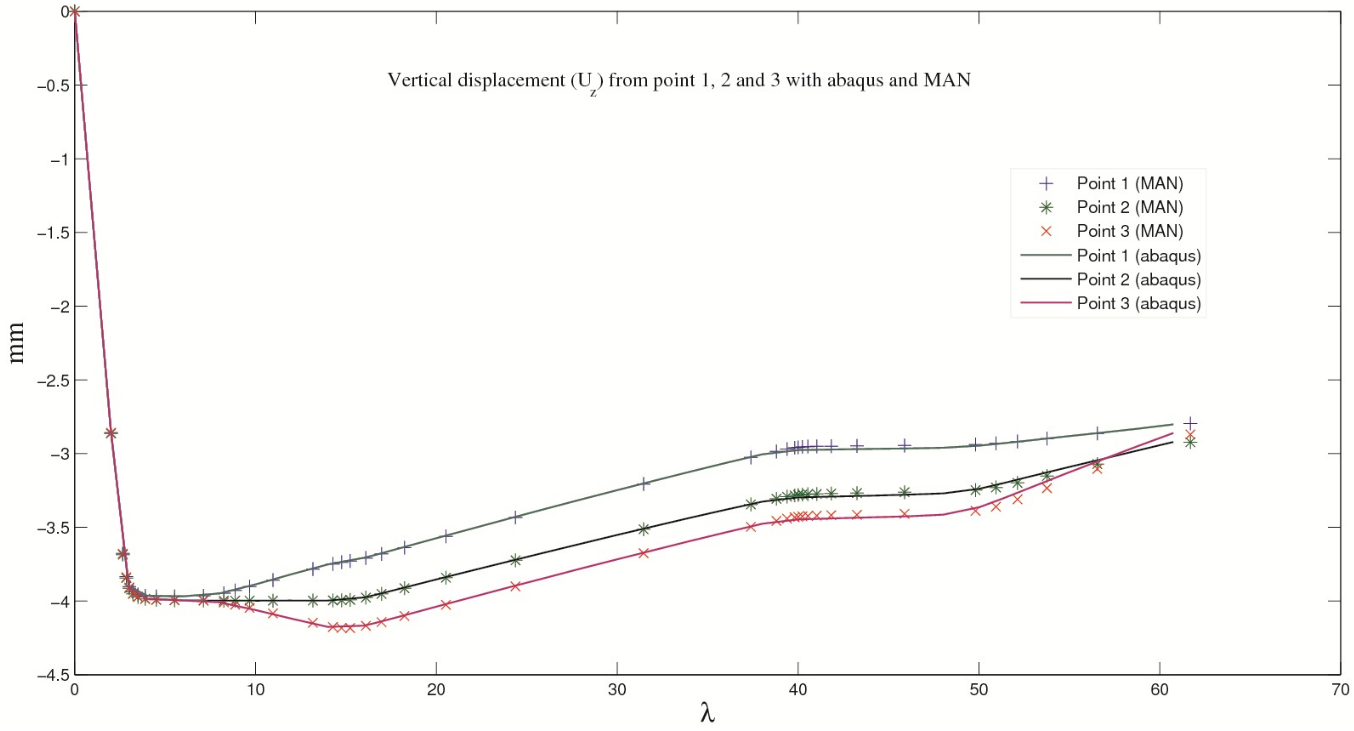

Figure 6: Vertical displacement profile along a conical cylinder. ANM versus Abaqus results.

rad. Note that for the projection equation (37), the parameter l_1 equals $\frac{l_t}{2}$. The parameter ϵ equals 23 as in the previous case.

The upper long edge of the strip is free from any boundary condition. The lower long edge uses symmetric boundary conditions: the model is constrained in the Y-translation. One of the short edges uses symmetric boundary conditions: the model is constrained in the X-translation. To avoid strip wandering during the conveying process, the second short edge is submitted to an uniformly distributed tensile load λF, where λ is the load parameter and the nominal force F equals 1 MPa. The material is elastic, linear and isotropic with data $E = 210$ MPa, $\nu = 0.3$.

The contact law is the same used in the previous validation cases: regularized law (15) with a stiffness parameter $\alpha = 10^6$ MPa.m and an adimensional regularisation parameter $\eta = 0.1$.

The deformable body is discretized using 256 8-noded quadrilateral shell elements with quadratic shape functions mentioned in section 3.3.

The non convex profile roll leads, as explained in [48], to buckling which is sensitive to the profile geometric parameters: the table length l_t and the angle p. This problem is simulated with 28 ANM steps using the ANM order equals 20 and the accuracy parameters $\delta = 10^{-6}$.

Figure 8 shows the deformed strip for a tensile load λF equal to 40 MPa. On this figure, the vertical displacement on the free part of the strip was increased fortyfold to make the buckling phenomenon visible on the figure.

In Figure 9 one sees the vertical displacement at the position $x = 470$ mm for different values of λF. For weak value of λF (i.e 8 MPa) the strip is slightly curved. As λF exceeds 13.1 MPa the buckling appears. For larger values of λF (i.e 44 MPa and 120 MPa), the buckling is well established but the distortion remains small. The strip profile and the start of the buckling are in good agreement with the results obtained by Gueydan [48] where the buckling initiates at λF=12.9 MPa.

4.2.2 Towards the simulation of irreversible wrinkling in industrial processes

In [1], sheet wrinkling in continuous annealing furnace was predicted in two steps by using a standard finite element code. In the first step, the combination of a traction load and of a roll profile led to an elastic buckling phenomenon. In a second step, a strip displacement was prescribed and the friction between roll and strip leads to an increase of the compressive stress and finally a wrinkle formation due to a plastic cumulative mechanism. We have established in the section 4.2.1 that the present solver is completely able to simulate the first step governed by geometric non-linearity and frictionless unilateral contact. The use of Automatic Differentiation was a crucial tool in a view to describe easily a

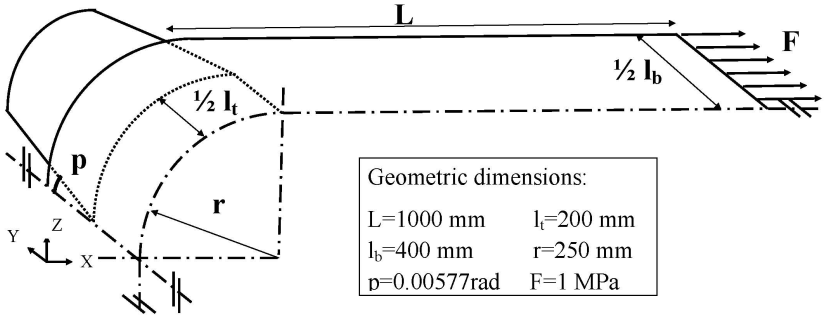

Figure 7: Industrial geometry

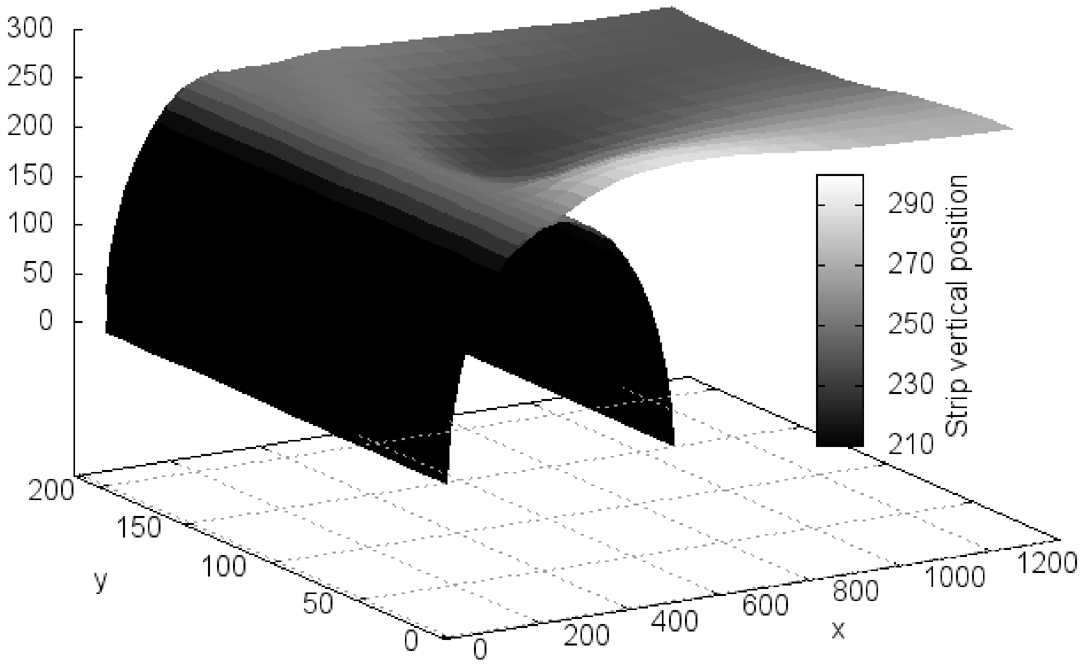

Figure 8: Zoom x40 on the strip buckling

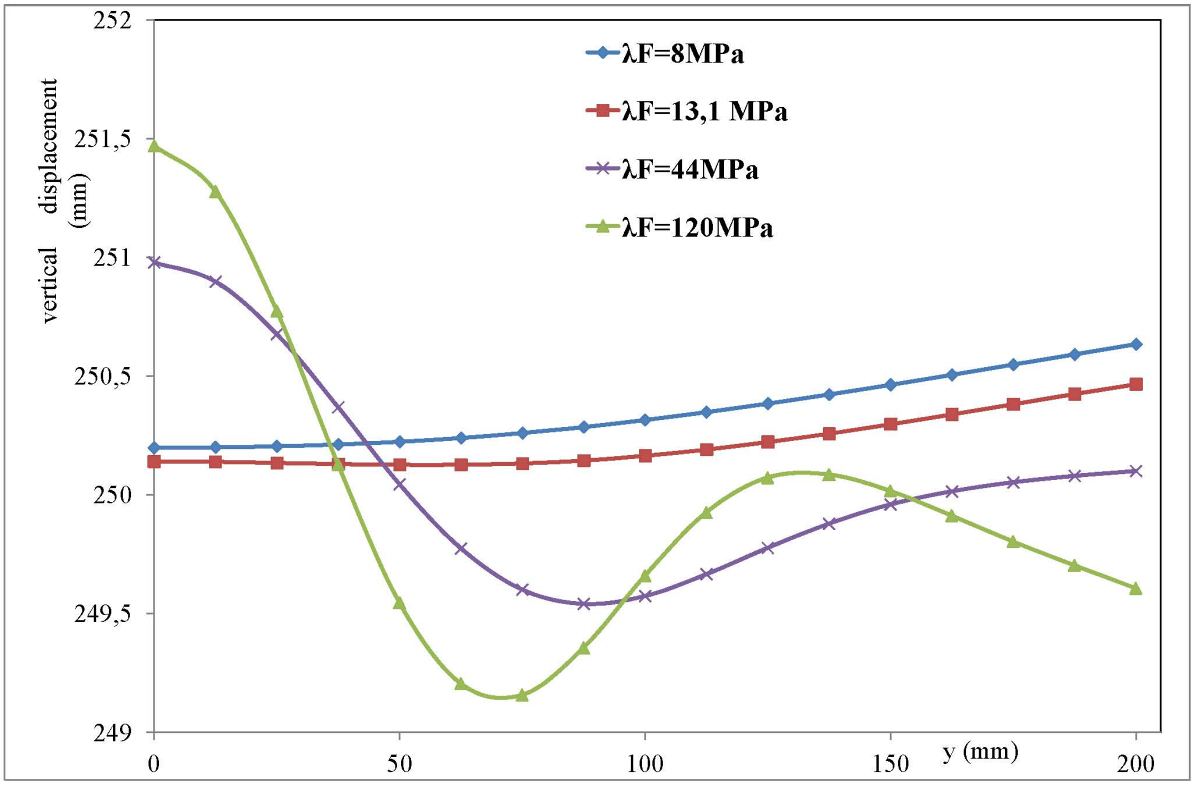

Figure 9: Vertical displacement observed at x=470 mm for different tensile load

general roll shape. In the same way, a general nonlinear constitutive law should be introduced, but we simply considered the linear Saint-Venant constitutive law (3) that is sufficient for a small strain buckling phenomenon. The relevance of the Saint-Venant constitutive law could be questioned, but not for such small strains. The framework of the present paper can be easily extended to hyperelasticity as in [43] [44] [45], but this will be insufficient for the aforementioned second step of [1]. For this second step, the plastic constitutive law and the friction law are not algebraic, but Differential Algebraic Equations (DAE). It is clear that the ideas of the present paper could also be applied to DAEs, but for simplicity we do not address this issue in this paper and we refer to [14] [19] for the treatment of plasticity and friction within ANM.

The small strain framework is sufficient in the study of strip conveying, but this is not the case for most of the forming processes that require large strain plasticity. Let us mention that ANM has been used with standard large strain plastic models [49]. So an extension of the present paper to DAEs would lead to a non-linear solver for many forming processes. The key point would be the automatic treatment of DAEs large strain plasticity.

5 Concluding remarks

In this paper, we presented an automatic approach to implement non-linear problems arising from continuum mechanics and other fields of engineering sciences. We focussed on problems including non-linear laws expressed with implicit and explicit functions. To achieve an automatic solver, an extension of the DIAMANT approach has been proposed. Then a library solving a wide range of non-linear analytic problems is available.

Moreover coupling our library with an automated Finite Element tool such as the Fenics project [50] could result in a real generic finite element solver and could be of great interest.

As applications, we solved non-linear unilateral contact problems modelled with Finite Element Method employing classical 3D or shell elements. This work is the first step to achieve the simulation of the strip conveying process. The next steps consist in accounting friction phenomena and including plasticity models.

This work has been supported by the French "Agence Nationale de la Recherche" (ANR) in the framework of the Project INSTABANDE.

References

[1] N. Jacques, A. Elias, M. Potier-Ferry, and H. Zahrouni. Buckling and wrinkling during strip conveying in processing lines. *Journal of Materials*

Processing Technology, 190:33–40, 2007.

[2] F.D. Fischer, F.G. Rammerstorfer, N. Friedl, and W. Wieser. Buckling phenomena related to rolling and levelling of sheet metal. *International Journal of Mechanical Sciences*, 42:1887–1910, 2000.

[3] P. Papadopoulos and J.M. Solberg. A lagrange multiplier method for the finite element solution of frictionless contact problems. *Mathematical and Computer Modelling*, 28:373 – 384, 1998.

[4] M. Fortin and R. Glowinski. *Augmented Lagrangian Methods: Applications to the Numerical Solutions of Boundary Value-Problems.* North-Holland, Amsterdam, 1992.

[5] J.C. Simo and T.A. Laursen. An augmented lagrangian treatment of contact problems involving friction. *Computers and Structures*, 42:97 – 116, 1992.

[6] N. Kikuchi. A smoothing technique for reduced integration penalty methods in contact problems. *International Journal for Numerical Methods in Engineering*, 18:343 – 350, 1982.

[7] D. Peric and D.R.J. Owen. Computational model for 3-d contact problems with friction based on the penalty method. *International Journal for Numerical Methods in Engineering*, 35:289 – 309, 1992.

[8] P. Wriggers. Finite element algorithms for contact problems. *Archives of Computational Methods in Engineering*, 2:1 – 49, 1995.

[9] M. C. Oliveira, J. L. Alves, and L. F. Menezes. Algorithms and strategies for treatment of large deformation frictional contact in the numerical simulation of deep drawing process. *Archives of Computational Methods in Engineering*, 15:113–162, 2008.

[10] A. Elhage-Hussein, N. Damil, and M. Potier-Ferry. An asymptotic numerical algorithm for frictionless contact problems. *Revue Européeenne des Eléments Finis*, 7:119–130, 1998.

[11] M. Potier-Ferry, N. Damil, B. Braikat, J. Descamps, J.M. Cadou, H.L. Cao, and A. Elhage-Hussein. Traitement des fortes non-linéarités par la méthode asymptotique-numérique. *Comptes Rendus de l'Académie des Sciences, Séries IIB*, 324:171 – 177, 1997.

[12] W. Aggoune, H. Zahrouni, and M. Potier-Ferry. High order prediction-correction algorithms for unilateral contact problems. *Journal of Computational and Applied Mathematics*, 168:1 – 9, 2004.

[13] W. Aggoune, H. Zahrouni, and M. Potier-Ferry. Asymptotic numerical methods for unilateral contact. *International Journal for Numerical Methods in Engineering*, 68:605 – 631, 2006.

[14] M. Assidi, H. Zahrouni, N. Damil, and M. Potier-Ferry. Regularization and perturbation technique to solve plasticity problems. *International Journal of Material Forming*, 2:1 – 14, 2009.

[15] H. Zahrouni, B. Braikat, N. Damil, and M. Potier-Ferry. Solving plasticity problems by a perturbation technique. *Proceedings in Applied Mathematics and Mechanics*, 5:455 – 456, 2005.

[16] H. Zahrouni, W. Aggoune, J. Brunelot, and M. Potier-Ferry. Asymptotic numerical method for strong nonlinearities. *Revue Européenne des Eléments Finis*, 13:97 –118, 2004.

[17] H. Abichou, H. Zahrouni, and M. Potier-Ferry. Asymptotic numerical method for problems coupling several nonlinearities. *Computer Methods in Applied Mechanics and Engineering*, 191:5795–5810, 2002.

[18] H. Zahrouni, B. Cochelin, and M. Potier-Ferry. Computing finite rotations of shells by an asymptotic-numerical method. *Computer Methods in Applied Mechanics and Engineering*, 175:71 – 85, 1999.

[19] F. Béchet, A. Lejeune, and M. Potier-Ferry. Taylor series to solve friction problems. *Comptes Rendus Mécanique*, 338:327 – 332, 2010.

[20] A Griewank and A Walther. On the efficient generation of taylor expansions for dae solutions by automatic differentiation. In J. Dongarra, K. Madsen, and J. Wasniewski, editors, *Applied Parallel Computing: State of the Art in Scientific Computing*, Lecture Notes in Computer Science 3732, pages 1089–1098. Springer-Verlag Berlin, 2006. 7th International Workshop on State of the Art in Scientific Computing, Lyngby, Denmark, Jun 20-23, 2004.

[21] I. Charpentier, A. Lejeune, and M. Potier-Ferry. The diamant approach for an efficient automatic differentiation of the asymptotic numerical method. In C. H. Bischof, H. M. Bücker, P. D. Hovland, U. Naumann, and J. Utke, editors, *Advances in Automatic Differentiation*, pages 139–149. Springer, 2008.

[22] J. Korelc. Automation of primal and sensitivity analysis of transient coupled problems. *Computational Mechanics*, 44:631–649, 2009.

[23] E. Tijskens, D. Roose, H. Ramon, and J. De Baerdemaeker. FastDer++, efficient automatic differentiation for non-linear PDE solvers. *Mathematics and computers in simulation*, 65:177–190, 2004. IMACS/IFAC Fourth International Symposium on Mathematical Modelling and Simulation in Agricultural and Bio-Industries, Haifa, Israel, june 12-14, 2001.

[24] E. Tijskens, D. Roose, H. Ramon, and J. De Baerdemaeker. Automatic differentiation for solving nonlinear partial differential equations: An efficient operator overloading approach. *Numerical Algorithms*, 30:259–301, 2002.

[25] R. A. Bartlett, D. M. Gay, and E. T. Phipps. Automatic differentiation of C++ codes for large-scale scientific computing. In V. N. Alexandrov, G. D. van Albada, P. M. A. Sloot, and J. Dongarra, editors, *Computational Science – ICCS 2006*, Lecture Notes in Computer Science 3994, pages 525–532, Heidelberg, 2006. Springer.

[26] R. A. Bartlett, B. G. Van Bloemen Waanders, and M. Berggren. Hybrid differentiation strategies for simulation and analysis of applications in C++. *ACM Transactions on Mathematical Software*, 35, 2008.

[27] Y. Koutsawa, I. Charpentier, E. M. Daya, and M. Cherkaoui. A generic approach for the solution of nonlinear residual equations. part i: The diamant toolbox. *Computer Methods in Applied Mechanics and Engineering*, 198:572 – 577, 2008.

[28] M. Bilasse, I. Charpentier, E.M. Daya, and Y. Koutsawa. A generic approach for the solution of nonlinear residual equations. part ii: Homotopy and complex nonlinear eigenvalue method. *Computer Methods in Applied Mechanics and Engineering*, 198:3999 – 4004, 2009.

[29] Arnaud Lejeune, Fabien Bchet, Hakim Boudaoud, Norman Mathieu, and Michel Potier-Ferry. Object-oriented design to automate a high order nonlinear solver based on asymptotic numerical method. *Advances in Engineering Software*, 48(0):70 – 88, 2012.

[30] G. Zavarise, P. Wriggers, E. Stein, and B. Schrefler. Real contact mechanisms and finite element formulation: a coupled thermomechanical approach. *International Journal for Numerical Methods in Engineering*, 35:767 – 785, 1992.

[31] A. Elhage-Hussein, M. Potier-Ferry, and N. Damil. A numerical continuation method based on padé approximants. *International Journal of Solids and Structures*, 37:6981–7001, 2000.

[32] I. Charpentier and M. Potier-Ferry. Différentiation automatique de la méthode asymptotique numérique typée : l'approche diamant. *Comptes Rendus Mécanique*, 336:336 – 340, 2008.

[33] Andrea Walther and Andreas Griewank. *A Package for the Automatic Differentiation of Algorithms Written in C/C++ -Version 2.1.12-stable*, 2010.

[34] À. Jorba and Z. Maorong. A software package for the numerical integration of odes by means of high-order taylor methods. *Experimental Mathematics*, 14:99–117, 2005.

[35] C. Bendtsen and O. Stauning. TADIFF, a flexible C++ package for automatic differentiation using taylor series expansion. Technical Report IMM–REP–1997–07, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, april 1997.

[36] M. C. Bartholomew-Biggs. Using forward accumulation for automatic differentiation of implicitly-defined functions. *Computational Optimization and Applications*, 9:65–84, 1998.

[37] C. H. Bischof and H. M. Bücker. Computing derivatives of computer programs. In J. Grotendorst, editor, *Modern Methods and Algorithms of Quantum Chemistry: Proceedings, Second Edition*, volume 3 of *NIC Series*, pages 315–327. NIC-Directors, Jul, 2000.

[38] M. E. Jerrell. Function minimization and automatic differentiation using c++. *SIGPLAN Not.*, 24:169–173, 1989.

[39] B. Stroustrup. *The C++ Programming Language.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2000.

[40] M. E. Jerrell. Automatic differentiation using almost any language. *SIGNUM Newsl.*, 24(1):2–10, 1989.

[41] C. H. Bischof, P. D. Hovland, and B. Norris. On the implementation of automatic differentiation tools. *Higher Order Symbolic Computation*, 21:311–331, 2008.

[42] L. B. Rall. *Automatic Differentiation: Techniques and Applications.* Lecture Notes in Computer Science 120. Springer, Berlin, 1981.

[43] I. Galliet. *Une version parallèle des méthodes asymptotiques numériques: Application à des structures complexes à base d'élastoméres.* PhD thesis, Université de Marseille II, France, 2000.

[44] S. Niroomandi, I. Alfaro, E. Cueto, and F. Chinesta. Model order reduction for hyperelastic materials. *International Journal for Numerical Methods in Engineering*, 81:1180–1206, 2009.

[45] S. Nezamabadi, H. Zahrouni, and J. Yvonnet. Solving hyperelastic material problems by asymptotic numerical method. *Computational Mechanics*, 47:77–92, 2011.

[46] N. Büchter, E. Ramm, and D; Roehl. Three-dimensional extension of nonlinear shell formulation based on the enhanced assumed strain concept. *International Journal for Numerical Methods in Engineering*, 37:2551–2568, 1994.

[47] J.C. Simo and M.S. Rifai. A class of mixed assumed strain methods and the method of incompatible modes. *International Journal for Numerical Methods in Engineering*, 29:1595–1638, 1990.

[48] V. Gueydan. *Modélisation numérique du flambage de bandes en acier dans les recuits continus.* PhD thesis, Université de Metz, France, 1997.

[49] M. Assidi. *Méthode asymptotique numérique pour la plasticité.* PhD thesis, Université de Metz, France, 2007.

[50] A. Logg. Automating the finite element method. *Archives of Computational Methods in Engineering*, 14:93–138, 2007.